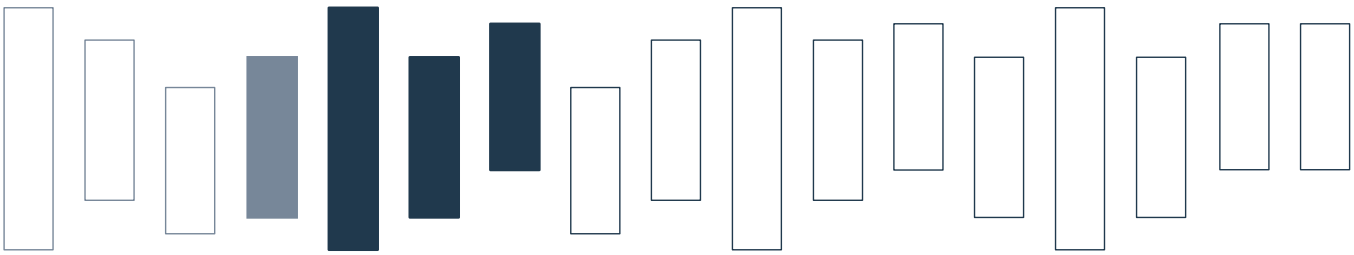
Technical Report AC-TR-21-007

April 2021

The Parameterized Complexity of Clustering Incomplete Data

Eduard Eiben, Robert Ganian, Iyad Kanj,
Sebastian Ordyniak, Stefan Szeider



This is the authors' copy of a paper that will appear in the proceedings of AAAI21, the the Thirty-Fifth AAAI Conference on Artificial Intelligence

www.ac.tuwien.ac.at/tr

The Parameterized Complexity of Clustering Incomplete Data

Eduard Eiben,[1] **Robert Ganian,**[2] **Iyad Kanj,**[3] **Sebastian Ordyniak,**[4] **Stefan Szeider**[2]

[1]Department of Computer Science, Royal Holloway, University of London, Egham, UK
[2]Algorithms and Complexity Group, TU Wien, Vienna, Austria
[3]School of Computing, DePaul University, Chicago, USA
[4]University of Leeds, School of Computing, Leeds, UK
{eduard.eiben,rganian}@gmail.com, ikanj@cdm.depaul.edu, sordyniak@gmail.com, stefan@szeider.net

Abstract

We study fundamental clustering problems for incomplete data. Specifically, given a set of incomplete d-dimensional vectors (representing rows of a matrix), the goal is to complete the missing vector entries in a way that admits a partitioning of the vectors into at most k clusters with radius or diameter at most r. We give tight characterizations of the parameterized complexity of these problems with respect to the parameters k, r, and the minimum number of rows and columns needed to cover all the missing entries. We show that the considered problems are fixed-parameter tractable when parameterized by the three parameters combined, and that dropping any of the three parameters results in parameterized intractability. A byproduct of our results is that, for the complete data setting, all problems under consideration are fixed-parameter tractable parameterized by $k + r$.

Introduction

We study fundamental clustering problems for incomplete data. In this setting, we are given a set of d-dimensional Boolean vectors (regarded as rows of a matrix), some of whose entries might be missing. The objective is to complete the missing entries in order to enable a "clustering" of the d-dimensional vectors such that elements in the same cluster are "similar."

There is a wealth of research on data completion problems (Candès and Plan 2010; Candès and Recht 2009; Candès and Tao 2010; Elhamifar and Vidal 2013; Ganian et al. 2018; Hardt et al. 2014) due to their ubiquitous applications in recommender systems, machine learning, sensing, computer vision, data science, and predictive analytics, among others. In these areas, data completion problems naturally arise after observing a sample from the set of vectors, and attempting to recover the missing entries with the goal of optimizing certain criteria. Some of these criteria include minimizing the number of clusters into which the completed vectors can be partitioned, or forming a large cluster, where the definition of what constitutes a cluster varies from one application to another (Balzano et al. 2012; Elhamifar 2016; Elhamifar and Vidal 2013; Yi et al. 2012). Needless to say, the clustering problem itself (*i.e.*, for complete data) is a fundamental problem whose applications span several areas of computing, including data mining, machine learning, pattern recognition, and recommender systems (Aggarwal and Reddy 2013; Gan, Ma, and Wu 2007; Leskovec, Rajaraman, and Ullman 2014; Mirkin 2005).

In many cases, the goal of clustering is to optimize the number of clusters and/or the degree of similarity within a cluster (*intra-cluster similarity*). To measure the intra-cluster similarity, apart from using an aggregate measure (*e.g.*, the variance in k-means clustering), two measures that have been studied use the *radius* (maximum distance to a selected "center" vector) and *diameter* (maximum distance between any two cluster-vectors) of the cluster (Charikar and Panigrahy 2004; Dyer and Frieze 1985; Feder and Greene 1988; Gąsieniec, Jansson, and Lingas 1999, 2004; Gonzalez 1985; Gramm, Niedermeier, and Rossmanith 2003). The radius is computed either with respect to a vector in the cluster itself or an arbitrary d-dimensional vector (Leskovec, Rajaraman, and Ullman 2014).

Regardless of which of the above measures of intra-cluster similarity is used, the vast majority of the clustering problems that arise are NP-hard. Consequently, heuristics are often used to cope with the hardness of clustering problems, trading in a suboptimal clustering for polynomial running time. In this paper we take a different approach: we maintain the optimality of the obtained clustering by relaxing the notion of tractability from polynomial-time to *fixed-parameter tractability* (FPT) (Cygan et al. 2015; Downey and Fellows 2013; Gottlob and Szeider 2008), where the running time is polynomial in the instance size but may involve a super-polynomial factor that depends only on some *problem parameter*, which is assumed to be small for certain instances of interest. In the context of clustering, two natural parameters that are desirable to be small are upper bounds on the number of clusters and the radius/diameter.

Contributions. Motivated by the above, we consider several fundamental clustering problems in the incomplete data setting. Namely, we consider the following three problems, referred to as IN-CLUSTERING-COMPLETION, ANY-CLUSTERING-COMPLETION, and DIAM-CLUSTERING-COMPLETION, that share a similar setting: In all three problems, the input is a (multi)set M of d-dimensional vectors over the Boolean domain[1], some of whose entries might be missing, and two parameters $r, k \in \mathbb{N}$; we use the symbol $\square$ to denote a missing vector entry. For IN-CLUSTERING-

[1]We view M as the (multi)set of rows of a Boolean matrix.

COMPLETION, the goal is decide whether the $\square$ entries in M can be completed to obtain a (complete) set of vectors M^* such that there is a subset $S \subseteq M^*$ with $|S| \leq k$ satisfying that, for every $\vec{a} \in M^*$, the Hamming distance between $\vec{a}$ and some vector in S is at most r. That is, the goal for IN-CLUSTERING-COMPLETION is to complete the missing entries so as to enable a partitioning of the resulting (complete) set into at most k clusters such that all vectors in the same cluster are within Hamming distance at most r from some "center" vector that belongs to the cluster itself. For ANY-CLUSTERING-COMPLETION, the goal is the same as that for IN-CLUSTERING-COMPLETION, except that the center vectors need not be in the set M (*i.e.*, are chosen from $\{0,1\}^d$). For DIAM-CLUSTERING-COMPLETION, the goal is to complete the missing entries in M so as to obtain a set M^* such that the vectors of M^* can be partitioned into at most k clusters/subsets, each of diameter at most r (where the diameter of a set of vectors S is the maximum pairwise Hamming distance over all pairs of vectors in the set). We denote by IN-CLUSTERING, ANY-CLUSTERING, and DIAM-CLUSTERING the complete versions of IN-CLUSTERING-COMPLETION, ANY-CLUSTERING-COMPLETION, and DIAM-CLUSTERING-COMPLETION, respectively; that is, the restrictions of the aforementioned data completion problems to input instances in which the set of vectors contains no missing entries.

Our first order of business is to obtain a detailed map of the parameterized complexity of the above three data completion problems. As we show in this paper, parameterization by $k + r$ is not sufficient to achieve tractability for any of these three problems: one needs to restrict the occurrences of the unknown entries in some way as well. We do so by adopting a third parameter defined as the minimum number of vectors and coordinates (or, equivalently, rows and columns in a matrix representation of M) needed to cover all the missing entries. This parameter, which we call the *covering number* or simply `cover`, is guaranteed to be small when the unknown entries arise from the addition of a small number of new rows and columns (*e.g.*, new users and attributes) into a known data-set; in particular, the parameter may be small even in instances with a large number of rows and columns that contain missing entries. The covering number has previously been used in the context of matrix completion (Ganian et al. 2018) and is in fact the least restrictive parameter considered in that paper.

Our main contribution is a complete parameterized complexity landscape for the complete and incomplete versions of all three clustering problems w.r.t. all combinations of the parameters k, r, and `cover`. Our main algorithmic contribution shows that the incomplete variants of all three clustering problems are fixed-parameter tractable parameterized by $k + r +$ `cover`, and as a consequence the complete variants are fixed-parameter tractable parameterized by $k + r$. Notably, our tractability results are obtained using *kernelization* (Fomin et al. 2019; Gaspers and Szeider 2014) and therefore provide efficient polynomial-time preprocessing procedures, which can be applied before the application of any available (even heuristic) clustering algorithm. To perform the kernelization, we apply a two-step approach: first we build on the well-known Sunflower Lemma (Erdös and Rado 1960) to develop new tools that allow us to reduce the number of rows in the target instance, and after that we use entirely different techniques to identify a small set of "distance-preserving" relevant coordinates. Together with a set of algorithmic lower bound results (and an XP algorithm for IN-CLUSTERING parameterized by k), this provides the comprehensive parameterized complexity landscape illustrated in Table 1. We also show that all our tractability results can be lifted from the Boolean domain to any finite domain, for the Hamming as well as Manhattan distance.

Related Work. In previous work, Hermelin and Rozenberg (2015) studied the CLOSEST STRING WITH WILDCARDS problem, which corresponds to ANY-CLUSTERING-COMPLETION with $k = 1$. Independently of our work, Koana et al. (2020b) very recently revisited the earlier work of Hermelin and Rozenberg (2015) and obtained, among other results, a fixed-parameter algorithm for that problem parameterized by r plus the maximum number of missing entries per row; in that same paper, they also studied IN-CLUSTERING-COMPLETION with $k = 1$. Even more recently, the same group (Koana, Froese, and Niedermeier 2020a) also studied a problem related to DIAM-CLUSTERING-COMPLETION for a single cluster, *i.e.*, for $k = 1$. They obtain a classification orthogonal to ours w.r.t. constant lower and upper bounds on the diameter and the maximum number of missing entries per row.

The main differences between the problems studied by Koana et al. (2020b; 2020a) and the restrictions of ANY-CLUSTERING-COMPLETION and IN-CLUSTERING-COMPLETION (studied in this paper) to $k = 1$ (*i.e.*, the restriction to the special case where we seek precisely 1 cluster) is the parameter used to capture the number of missing entries per row. Indeed, the authors of these works consider the maximum number of missing entries (over all rows), whereas we consider the parameter `cover`. The two parameters are orthogonal: there are instances in which the maximum number of missing entries per row is very small yet `cover` is large, and vice versa.

The parameterized complexity of a related problem—MATRIX COMPLETION—has been studied in a different context than that of clustering (Ganian et al. 2018); the problem considered therein corresponds to the special case of IN-CLUSTERING-COMPLETION in which the clustering radius r is 0. There is also an extensive body of research on clustering problems for complete data. Examples include the work of Frances and Litman (1997), Gramm, Niedermeier and Rossmanith (2003), as well as many other works (Boucher and Ma 2011; Cabello et al. 2011; Fomin et al. 2020; Fomin, Golovach, and Panolan 2020; Fomin, Golovach, and Simonov 2019; Gąsieniec, Jansson, and Lingas 1999, 2004; Gonzalez 1985). Note also that IN-CLUSTERING and ANY-CLUSTERING are special instances of the well-known k-center problem.

We remark that related problems have also been studied by a variety of other authors, such as, e.g., Chen, Hermelin, Sorge (2019).

Parameter:	k	r	$k+r$	$k+r+\texttt{cover}$
IN-CLUSTERING	W[2]-c	paraNP-c	FPT	N/A
ANY/DIAM-CLUSTERING	paraNP-c	paraNP-c	FPT	N/A
IN/ANY/DIAM-CLUSTERING-C	paraNP-c	paraNP-c	paraNP-c	FPT

Table 1: Parameterized complexity results for exact clustering with complete data (top) and incomplete data (bottom). FPT means fixed-parameter tractability, while paraNP-c and W[2]-c mean completeness for these complexity classes and indicate fixed-parameter intractability.

Preliminaries

A *parameterized problem* Q is a subset of $\Omega^* \times \mathbb{N}$, where Ω is a fixed alphabet. Each instance of Q is a pair (I, κ), where $\kappa \in \mathbb{N}$ is called the *parameter*. A parameterized problem Q is *fixed-parameter tractable* if there is an algorithm, called an FPT-*algorithm*, that decides whether an input (I, κ) is a member of Q in time $f(\kappa) \cdot |I|^{\mathcal{O}(1)}$, where f is a computable function and $|I|$ is the input instance size. The class FPT denotes the class of all fixed-parameter tractable parameterized problems. A parameterized problem is *kernelizable* if there exists a polynomial-time reduction that maps an instance (I, κ) of the problem to another instance (I', κ') such that (1) $|I'| \leq f(\kappa)$ and $\kappa' \leq f(\kappa)$, where f is a computable function, and (2) (I, κ) is a YES-instance of the problem if and only if (I', κ') is. The instance (I', κ') is called the *kernel* of I. It is well known that a decidable problem is FPT if and only if it is kernelizable. A hierarchy, the W-hierarchy, of parameterized complexity has been defined, and the notions of hardness and completeness have been introduced for each level W[i] of the W-hierarchy for $i \geq 1$. It is commonly believed that W[2] $\supset$ W[1] $\supset$ FPT, and the notion of W[1]-hardness has served as the main working hypothesis of fixed-parameter intractability. An even stronger notion of intractability is that of paraNP-hardness, which contains all parameterized problems which remain NP-hard even if the parameter is fixed to a constant. We refer readers to the relevant literature (Flum and Grohe 2006; Downey and Fellows 2013; Cygan et al. 2015) for more information.

Let $\vec{a}$ and $\vec{b}$ be two binary vectors. We denote by $\Delta(\vec{a}, \vec{b})$ the set of coordinates in which $\vec{a}$ and $\vec{b}$ are guaranteed to differ, *i.e.*, $\Delta(\vec{a}, \vec{b}) = \{\, i \mid (\vec{a}[i] = 1 \wedge \vec{b}[i] = 0) \vee (\vec{a}[i] = 0 \wedge \vec{b}[i] = 1) \,\}$, and we denote by $\delta(\vec{a}, \vec{b})$ the *Hamming distance* between $\vec{a}$ and $\vec{b}$ measured only between known entries, *i.e.*, $|\Delta(\vec{a}, \vec{b})|$. We denote by $\Delta(\vec{a})$ the set $\Delta(\vec{0}, \vec{a})$, and for a set C of coordinates, we denote by $\Delta^{-1}(C)$ the vector that is 1 at precisely the coordinates in C and 0 at all other coordinates. We extend this notation to sets of vectors and a family of coordinate sets, respectively. For a set N of vectors in $\{0,1\}^d$ and a family $\mathcal{C}$ of coordinate sets, we denote by $\Delta(N)$ the set $\{\, \Delta(\vec{v}) \mid \vec{v} \in N \,\}$ and by $\Delta^{-1}(\mathcal{C})$ the set $\{\, \Delta^{-1}(C) \mid C \in \mathcal{C} \,\}$. We say that a vector $\vec{a} \in \{0,1\}^d$ is a t-*vector* if $|\Delta(\vec{a})| = t$ and we say that $\vec{a}$ *contains* a subset S of coordinates if $S \subseteq \Delta(\vec{a})$. For a subset $S \subseteq \{0,1\}^d$ and a vector $\vec{a} \in \{0,1\}^d$, we denote by $\delta(S, \vec{a})$ the minimum Hamming distance between $\vec{a}$ and the vectors in S, *i.e.*, $\delta(S, \vec{a}) = \min_{\vec{s} \in S} \delta(\vec{s}, \vec{a})$. We denote by $\gamma(S)$ the *diameter* of S, *i.e.*, $\gamma(S) := \max_{\vec{s_1}, \vec{s_2} \in S} \delta(\vec{s_1}, \vec{s_2})$.

Let $M \subseteq \{0,1\}^d$ and let $[d] = \{1, \ldots, d\}$. For a vector $\vec{a} \in M$, we denote by $N_r(\vec{a})$ the r-*Hamming neighborhood* of $\vec{a}$, *i.e.*, the set $\{\, \vec{b} \in M \mid \delta(\vec{a}, \vec{b}) \leq r \,\}$ and by $N_r(M)$ the set $\bigcup_{\vec{a} \in M} N_r(\vec{a})$. Similarly, we denote by $N_{=r}(\vec{a})$ the the set $\{\, \vec{b} \in M \mid \delta(\vec{a}, \vec{b}) = r \,\}$ and by $N_{=r}(M)$ the set $\bigcup_{\vec{a} \in M} N_{=r}(\vec{a})$. We say that $M^* \subseteq \{0,1\}^d$ is a *completion* of $M \subseteq \{0,1,\square\}^d$ if there is a bijection $\alpha : M \to M^*$ such that for all $\vec{a} \in M$ and all $i \in [d]$ it holds that either $\vec{a}[i] = \square$ or $\alpha(\vec{a})[i] = \vec{a}[i]$.

Let $\{\vec{v}_1, \ldots, \vec{v}_n\}$ be an arbitrary but fixed ordering of a subset M of $\{0,1,\square\}^d$. If $\vec{v}_i[j] = \square$, we say that $\square$ at $\vec{v}_i[j]$ is *covered* by row i and column j. The *covering number* of M, denoted as $\texttt{cover}(M)$ or simply $\texttt{cover}$, is the minimum value of $r+c$ such that there exist r rows and c columns in M with the property that each occurrence of $\square$ is covered by one of these rows or columns. We will generally assume that for a set $M \in \{0,1,\square\}^d$ we have computed sets T_M and R_M such that $\texttt{cover}(M) = |T_M| + |R_M|$ and each $\square$ occurring in a vector $\vec{v} \in M$ is covered by a row in R_M or a column in T_M; we note that this computation can be done in polynomial time (Ganian et al. 2018, Proposition 1), and in our algorithms parameterized by $\texttt{cover}(M)$, we will generally assume that T_M and R_M have already been pre-computed.

It will sometimes be useful to argue using the *compatibility graph* G associated with an instance $\mathcal{I}$ of IN/ANY/DIAM-CLUSTERING-COMPLETION. This is the graph whose vertex set is M and which has an edge between two vectors $\vec{a}$ and $\vec{b}$ if and only if: $\delta(\vec{a}, \vec{b}) \leq r$ (for the IN- and DIAM- variants) or $\delta(\vec{a}, \vec{b}) \leq 2r$ (for the ANY- variant). Notice that vectors in different connected components of G cannot interact with each other: every cluster containing vectors from one connected component cannot contain a vector from any other connected component.

The Toolkit

In this section, we present key structural results that are employed in several algorithms and lower bounds in the paper. The first part of our toolkit and structural results for matrices are obtained by exploiting the classical sunflower lemma of Erdös and Rado, a powerful combinatorial tool that has been used to obtain kernelization algorithms for many fundamental parameterized problems (Fomin et al. 2019). A *sunflower* in a set family $\mathcal{F}$ is a subset $\mathcal{F}' \subseteq \mathcal{F}$ such that all pairs of elements in $\mathcal{F}'$ have the same intersection.

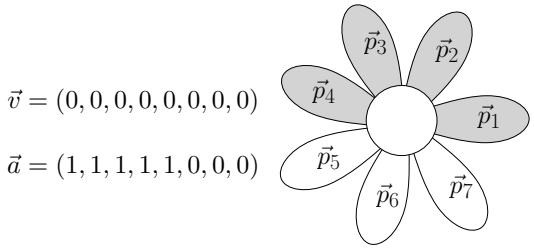


Figure 1: The figure shows an example for the setting in Lemma 3. Here $r = 3$ and $t = 2$ and the figure shows the vectors $\vec{v}$ and $\vec{a}$ as well as the sunflower resulting from the vectors $\vec{p}_1, \ldots, \vec{p}_7$ with $\vec{p}_i[j] = 1$ if and only if either $j = 1$ or $j = i + 1$. In this example three of the petals, *i.e.*, the white petals $\vec{p}_5$, $\vec{p}_6$, and $\vec{p}_7$, only share the core of the sunflower with $\vec{a}$, which implies that all three of these petals are of maximum Hamming distance to $\vec{a}$.

Lemma 1 (Erdös and Rado 1960; Flum and Grohe 2006)**.** *Let $\mathcal{F}$ be a family of subsets of a universe U, each of cardinality exactly b, and let $a \in \mathbb{N}$. If $|\mathcal{F}| \geq b!(a-1)^b$, then $\mathcal{F}$ contains a sunflower $\mathcal{F}'$ of cardinality at least a. Moreover, $\mathcal{F}'$ can be computed in time polynomial in $|\mathcal{F}|$.*

Finding Irrelevant Vectors. The first structural lemma we introduce is Lemma 3, which is also illustrated in Figure 1. Intuitively speaking, the lemma says that if the t-Hamming neighborhood of a vector $\vec{v}$ contains a large sunflower, then at least one of its elements can be removed without changing the maximum distance to any vector $\vec{a}$ that is of distance at most r to the elements in the sunflower. The proof of Lemma 3 utilizes the straightforward Lemma 2, which captures a useful observation that is also used in other proofs.

We note that the idea of applying the Sunflower Lemma on a similar set representation of an instance was also used in a previous work by Marx (2005) (see also Kratsch, Marx and Wahlström, 2016) to obtain FPT and kernelization results, albeit in the context of studying the weighted satisfiability of CSPs. There, the authors used the sunflower to reduce the arity of constraints by replacing the sets in the sunflower (which correspond to the scope of the constraints) by constraints defined over the petals without the core plus one additional constraint defined only on the variables of the core. We, however, use the sunflower in a different manner, namely to identify irrelevant vectors that can be safely removed from the instance. Note also that in contrast to many other applications of the sunflower, where all petals are removed and replaced by the core, this is not possible in our setting since we need to keep a certain number of petals in order to maintain the clustering properties of the instance.

Lemma 2. *Let $t, r \in \mathbb{N}$. Let $N \subseteq \{0,1\}^d$ be a set of t-vectors such that $\mathcal{F} := \Delta(N)$ is a sunflower with core C. If $|N| > r$, then for every vector $\vec{a} \in \{0,1\}^d$ with $|\Delta(\vec{a})| \leq r$, N contains a vector that has maximum distance to $\vec{a}$ among all t-vectors that contain C.*

Lemma 3. *Let $t, r \in \mathbb{N}$, $\vec{v} \in \{0,1\}^d$, $N \subseteq N_{=t}(\vec{v})$, and $\mathcal{F} := \{\, \Delta(\vec{v}, \vec{x}) \mid x \in N \,\}$ such that $\mathcal{F}$ is a sunflower with core C. Then for every subset N' of N with $|N'| \geq r+t+2$ and every vector $\vec{a} \in \{0,1\}^d$ such that $\delta(N', \vec{a}) \leq r$, we have $\delta(\vec{f}, \vec{a}) \leq \max_{\vec{x} \in N' \setminus \{\vec{f}\}} \delta(\vec{x}, \vec{a})$ for every $\vec{f} \in N$.*

The following lemma now employs Lemmas 3 and 1 to show that if the t-Hamming neighborhood of a vector $\vec{v}$ is large enough, at least one of its elements can be removed without changing the clustering properties of the instance.

Lemma 4. *Let $k, r, t \in \mathbb{N}$, $M \subseteq \{0,1\}^d$, $\vec{v} \in M$, and $N := N_{=t}(\vec{v}) \cap M$. If $|N| \geq t!(k(r+t+2))^t$, then there is a vector $\vec{f} \in N$ satisfying the following two properties:* **(P1)** *for every set $S \subseteq \{0,1\}^d$ with $|S| \leq k$ and satisfying $\delta(S, \vec{m}) \leq r$ for every $\vec{m} \in M$, it holds that $\max_{\vec{y} \in M} \delta(S, \vec{y}) = \max_{\vec{y} \in M \setminus \{\vec{f}\}} \delta(S, \vec{y})$; and* **(P2)** *$M$ has a partition into at most k clusters, each of diameter at most r, if and only if $M \setminus \{\vec{f}\}$ does. Moreover, $\vec{f}$ can be determined in time polynomial in M.*

Irrelevant Coordinates and Diameter Bound. Our clustering algorithms for IN/ANY/DIAM-CLUSTERING-COMPLETION will broadly proceed in two steps. Given an instance $\mathcal{I} = (M, k, r)$ of IN/ANY/DIAM-CLUSTERING-COMPLETION, we will first compute an equivalent instance (M', k, r) such that the size of M' can be bounded by a function of the parameter $k + r + \texttt{cover}(M)$ (this is done by the *irrelevant vector technique*). However, since our aim is to obtain a kernel, we then still need to reduce the number of coordinates for every vector in M'. That is where we use our *irrelevant coordinate technique*. This subsection introduces the tools and notions that are central to this technique. Throughout this section, we will assume that $\mathcal{I} = (M, k, r)$ is the considered input instance of IN/ANY/DIAM-CLUSTERING-COMPLETION.

Let $Z(M)$ for $M \subseteq \{0,1\}^d$ be the set of all coordinates i such that at least two vectors in M disagree on their i-th coordinate, *i.e.*, there are two vectors $\vec{y}, \vec{y}' \in M$ such that $\{\vec{y}[i], \vec{y}'[i]\} = \{0,1\}$. Intuitively, $Z(M)$ is the set of *important coordinates*, since all other coordinates can be safely removed from the instance; this is because they can always be completed to the same value and hence do not influence the properties of a clustering of M. Note that if we could show that the number of important coordinates is bounded by a function of M' and our parameter $k + r + \texttt{cover}(M)$, then we would obtain a kernel by simply removing all coordinates that are not important. Unfortunately, this is not the case for two reasons: First the compatibility graph $G(\mathcal{I})$ can consist of more than one component and the vectors in different components can differ in arbitrary coordinates. Furthermore, even inside a component the number of important coordinates can be arbitrary large. For instance, a component could consist of the all-zero vector, the all-one vector, and the all $\square$ vector. Note that the all $\square$ vector is crucial for this example and indeed, the next lemma shows that if we restrict ourselves to a component containing only vectors in $M \setminus R_M$, then the number of important coordinates can be bounded in terms of the diameter and the number of vectors inside the component.

Lemma 5. *Let $M' \subseteq M \setminus R_M$ such that $G(\mathcal{I})[M']$ is connected. Then $|Z(M') \setminus T_M| \leq \gamma(M')(|M'| - 1)$.*

The next lemma now shows how to bound the diameter of every component in $M \setminus R_M$ in terms of our parameter $k + r + \texttt{cover}(M)$.

Lemma 6. *Let $\mathcal{I} = (M, k, r)$ be an instance of* IN-CLUSTERING-COMPLETION*,* ANY-CLUSTERING-COMPLETION*, or* DIAM-CLUSTERING-COMPLETION *and let $M' \subseteq M \setminus R_M$ be such that $G(\mathcal{I})[M']$ is connected. Then $\mathcal{I}$ is a* NO*-instance if either:*

- *$\mathcal{I}$ is an instance of* IN-CLUSTERING-COMPLETION *and $\gamma(M') > 3rk - r + |T_M|$;*
- *$\mathcal{I}$ is an instance of* ANY-CLUSTERING-COMPLETION *and $\gamma(M') > 4rk - r + |T_M|$; or*
- *$\mathcal{I}$ is an instance of* DIAM-CLUSTERING-COMPLETION *and $\gamma(M') > 2rk - r + |T_M|$.*

We now already know how to bound the number of important coordinates inside a component of $M \setminus R_M$. Unfortunately, as we have illustrated previously, it is not possible to do the same for $M \setminus R_M$, let alone for the complete vector set M. However, the following lemma shows that there is a (small) set D' of coordinates that satisfy a slightly weaker property: it preserves distances up to r within components of $M \setminus R_M$ as well as to and between the vectors in R_M.

Lemma 7. *Let $M' \subseteq M$ and r' be a natural number. Then there is a subset $D' \subseteq [d]$ of coordinates such that:*

- *(C1) $|D'| \leq (k\gamma_{\max}(M') + |R_M|(|M'| - 1))(r' + 1) + |T_M|$ and*
- *(C2) for any two vectors $\vec{m}$ and $\vec{m}'$ in M' such that $\vec{m}$ and $\vec{m}'$ are in the same component of $G(\mathcal{I})[M']$ or one of $\vec{m}$ or $\vec{m}'$ is in R_M, it holds that $\delta(\vec{m}, \vec{m}') = \delta(\vec{m}[D'], \vec{m}'[D'])$ if $\delta(\vec{m}, \vec{m}') \leq r'$ and $\delta(\vec{m}[D'], \vec{m}'[D']) > r'$, otherwise.*

Here $\gamma_{\max}(M')$ is equal to the maximum diameter of any connected component of $G(\mathcal{I})[M']$.

The following lemma now shows that keeping only the set D' of coordinates is sufficient to preserve the equivalence for our three clustering problems.

Lemma 8. *Let $M' \subseteq M$. Then we can compute a set $D' \subseteq [d]$ of coordinates in polynomial-time such that:*

- *$|D'| \leq (k\gamma_{\max}(M') + |R_M|(|M'| - 1))(2r + 1) + |T_M|$ and (M', k, r) is a* YES*-instance of* ANY-CLUSTERING-COMPLETION *if and only if $(M'_{D'}, k, r)$ is.*
- *$|D'| \leq (k\gamma_{\max}(M') + |R_M|(|M'| - 1))(r + 1) + |T_M|$ and for $X \in \{$*IN*,*DIAM*$\}$: (M', k, r) is a* YES*-instance of X-*CLUSTERING-COMPLETION *if and only if $(M'_{D'}, k, r)$ is.*

Here, $M'_{D'}$ is the matrix obtained from M' after removing all coordinates (columns) that are not in D'.

Clustering with Incomplete Data

We will show that IN/ANY/DIAM-CLUSTERING-COMPLETION are fixed-parameter tractable parameterized by $k + r + \texttt{cover}(M)$. Our algorithmic results are achieved via kernelization: we will apply the irrelevant vector and irrelevant coordinate techniques to obtain an equivalent instance of size upper bounded by a function of $k + r + \texttt{cover}(M)$.

Note that this implies that also the variants IN/ANY/DIAM-CLUSTERING for complete data are fixed-parameter tractable parameterized by only $k + r$ (and also have a polynomial kernel) and, as we will show in a later section, both parameters are indeed required. To explain how we obtained our results, we will start by considering the general procedures for complete data first and then provide the necessary changes for the case of incomplete data. Throughout the section we will assume that (M, k, r) is the given instance of IN/ANY/DIAM-CLUSTERING-COMPLETION. Recall that, when using the parameter $\texttt{cover}(M)$, we will use the sets T_M and R_M (as defined in the preliminaries), where $T_M \subset [d]$, $R_M \subset M$, and $|T_M| + |R_M| = \texttt{cover}(M)$, and such that all $\square$'s in $M \setminus R_M$ occur only in coordinates in T_M.

Informal description of the algorithm for complete data. To perform kernelization, we start by identifying and removing irrelevant vectors; those are vectors that can be removed from the instance and safely added back to any valid clustering of the reduced instance to yield a valid clustering of the original instance. One caveat is that, for IN-CLUSTERING, the removed vectors may serve as cluster centers, and hence, such vectors will have to be represented in the reduced instance; we will discuss later (below) how this issue is dealt with. To identify irrelevant vectors, we first show that, for each vector, we can compute a "representative set" of vectors of its ($\leq r$)-neighborhood whose size is upper bounded by a function of the parameter. The identification of representative sets is achieved via a non-trivial application of the Sunflower Lemma (and several other techniques) in Lemmas 3, 4 as well as Lemma 9 for ANY-CLUSTERING, Lemma 9 and 11 for IN-CLUSTERING, and Lemma 13 for ANY-CLUSTERING. The union of these representative sets yields a reduced instance whose number of vectors is upper bounded by a function of the parameter. For the final step of our algorithm we use our toolkit to reduce the number of dimensions for every vector in the reduced instance. This is already sufficient to solve ANY-CLUSTERING.

As for IN-CLUSTERING, we need to ensure that the centers of the clusters in any valid solution are represented in the reduced instance (whose size is now bounded by a function of the parameter). To do so, we partition the set of vectors removed from the reduced instance into equivalence classes based on their "trace" on the set of important coordinates; the number of equivalence classes is upper bounded by a function of the parameter. Since each potential center must be within distance r from some vector in the reduced instance, for each (irrelevant) vector $\vec{x}$ that differs in at most r important coordinates from some vector in the reduced instance, we add a vector from the equivalence class of $\vec{x}$ (that represents $\vec{x}$) whose distance to the vectors in the reduced instance w.r.t. nonimportant coordinates (which all vectors in the reduced instance agree on) is minimum. Lemma 11 provides a bound on the number of these added vectors.

Finding Redundancy when Data is Missing. In the case of incomplete data, we will in principle employ the same general strategy that we used for clustering problems with complete data. Namely, we will again identify irrelevant vectors

and coordinates whose removal results in an instance whose size can be bounded by our parameter. However, due to the presence of incomplete data, we need to make significant adaptations at every step of the algorithm.

Consider the first step of the algorithm, which allowed us to identify and remove irrelevant vectors. For this step, we can focus only on the vectors in $M \setminus R_M$, since $|R_M|$ is already bounded by $\texttt{cover}(M)$; crucially, this allows us to assume that vectors only have $\square$-entries at positions in T_M.

Now consider Lemma 4, which allowed us to remove any vector, say $\vec{f}$, in a sufficiently large sunflower occurring in the t-Hamming neighborhood of some vector $\vec{v}$. Informally, this was because in every solution of the reduced instance, a large part of the sunflower must end up together in one of the clusters; this in turn meant that for every vector in the cluster there is a vector in the sunflower that is at least as far as $\vec{f}$. This is what allowed us to argue that $\vec{f}$ can always be safely added back into that cluster. But this can no longer be guaranteed once $\square$-entries are allowed, since whether $\vec{f}$ can be added back into the cluster or not depends on how the other vectors in the sunflower have been completed.

Note that the problem above would disappear if we could ensure that a sufficiently large number of vectors from the initial sunflower that end up together in the same cluster have the $\square$-entries at the *exact same positions*. Since we observed earlier that we can assume that all vectors have their $\square$-entries only in T_M, and consequently there are at most $2^{|T_M|}$ different allocations of the $\square$-entries to these vectors, we can now enforce this by enlarging the initial sunflower by a factor of $2^{|T_M|}$. This approach allows us to obtain the following lemma, which uses Lemma 4 in a way that allows us to reduce the number of vectors for IN-CLUSTERING-COMPLETION and ANY-CLUSTERING-COMPLETION.

Lemma 9. *Let $k, r \in \mathbb{N}$ and $M \subseteq \{0,1,\square\}^d$. Then there is a subset M' of M with $R_M \subseteq M'$ satisfying:*

- *(P1) For every $\vec{v} \in M \setminus R_M$ it holds that $|N_r(\vec{v}) \cap M' \setminus R_M| \leq 2^{|T_M|}(\sum_{t=1}^{r} t!(k(r+t)+2)^t)$; and*
- *(P2) for every set $S \subseteq \{0,1\}^d$ with $|S| \leq k$ and satisfying $\delta(S, \vec{m}) \leq r$ for every $\vec{m} \in M$ it holds that $\max_{\vec{y} \in M} \delta(S, \vec{y}) = \max_{\vec{y} \in M'} \delta(S, \vec{y})$.*

Moreover, M' can be computed in time polynomial in M.

Using the above Lemma 9 together with our toolbox (for reducing the number of relevant coordinates), we are now ready to show our first fixed-parameter algorithm for ANY-CLUSTERING-COMPLETION.

Theorem 10. ANY-CLUSTERING-COMPLETION *is* FPT *parameterized by $k + r + \texttt{cover}(M)$.*

Towards showing our kernelization result for IN-CLUSTERING-COMPLETION, we need to add back some vectors that can be potential centers for the clusters containing vectors of M'. The main idea for the case of complete data is the observation that every vector in M that can act as a potential center for the instance on M' must be within the r-neighborhood of some vector in M' and moreover among all (potentially many vectors within the r-neighborhood of a vector in M'), we can chose any vector, which is closest w.r.t. the unimportant coordinates, *i.e.*, the coordinates in $[d] \setminus Z(M')$. This way the number of potential vectors that can act as a center for a vector in M' can be bounded by the parameter. For the case of incomplete data we need to consider an additional complication, namely, that the $\square$ entries of the vectors in M' (which can be changed without increasing the Hamming distance to the vector), can increase the size of the r-Hamming neighborhood of every such vector now significantly. For instance, the potential r-Hamming neighborhood of a vector in $M' \setminus R_M$ increases by a factor of $2^{|T_M|}$ and the potential r-Hamming neighborhood of a vector $\vec{x}$ in R_M can only be bounded by $2^{\gamma(M')(|M'|-1)}3^{|T_M|}$, since every important coordinate of $\vec{x}$ could be a $\square$.

Lemma 11. *Let (M, k, r) be an instance of* IN-CLUSTERING-COMPLETION *and $M' \subseteq M$ with $R_M \subseteq M'$. Then there is a set M'' with $M' \subseteq M'' \subseteq M$ of size at most $|M'| + 3^{|T_M|} r^{|R_M|} + k 3^{2|T_M|} r^{|R_M|} 2^{\gamma_{\max}(M')(|M'|-1)}$ such that there is a set $S \subseteq M$ with $|S| \leq k$ satisfying $\max_{\vec{y} \in M'} \delta(S, \vec{y}) \leq r$ if and only if there is a set $S \subseteq M''$ with $|S| \leq k$ satisfying $\max_{\vec{y} \in M'} \delta(S, \vec{y}) \leq r$. Moreover, M'' can be computed in polynomial time.*

With Lemma 11 in hand, we can establish the fixed-parameter tractability of IN-CLUSTERING-COMPLETION.

Theorem 12. IN-CLUSTERING-COMPLETION *is* FPT *parameterized by $k + r + \texttt{cover}(M)$.*

We now proceed to the last of the three problems considered in this section, DIAM-CLUSTERING-COMPLETION. Apart from the issue that we already had for IN-CLUSTERING-COMPLETION and ANY-CLUSTERING-COMPLETION that we require a sunflower of vectors with all $\square$s in the same position, we now have the additional complication that we can no longer assume that the $\square$-entries of vectors that end up in the same cluster are completed in the same way; note that this is not an issue for IN-CLUSTERING-COMPLETION and ANY-CLUSTERING-COMPLETION since there one can always assume that all elements in a cluster are completed the same way as the center vector. We show that this problem can be handled by increasing the size of the sunflower by an additional factor of $2^{|T_M|}$. Because of the same issue, we also need to take into account the potential distance between different vectors in the same cluster arising from the possibility of different completions of the coordinates in T_M. This leads to the following version of Lemma 9 for DIAM-CLUSTERING-COMPLETION.

Lemma 13. *Let $k, r \in \mathbb{N}$, and $M \subseteq \{0,1,\square\}^d$. Then there is a subset M' of M with $R_M \subseteq M'$ satisfying:* **(P1)** *For every $\vec{v} \in M \setminus R_M$ it holds that $|N_r(\vec{v}) \cap (M' \setminus R_M)| \leq 2^{|T_M|}(\sum_{t=1}^{r+|T_M|} t!(2^{|T_M|}k(r+t+|T_M|)+2)^t)+1$; and* **(P2)** *$M$ has a completion with a partition into at most k clusters of diameter at most r if and only if M' does. Moreover, M' can be computed in time polynomial in M.*

We can now prove that DIAM-CLUSTERING-COMPLETION is FPT w.r.t. the three parameters.

Theorem 14. DIAM-CLUSTERING-COMPLETION *is* FPT *parameterized by $k + r + \texttt{cover}(M)$.*

Lower-Bound Results

This section is dedicated to showing that the parameterizations used in the algorithms presented up to this point are necessary to achieve (fixed-parameter) tractability. We do so by providing a number of hardness reductions.

It is known that ANY-CLUSTERING is NP-complete for $r = 2$ (Jiao, Xu, and Li 2004). Our first two hardness results show that the other two clustering problems also NP-complete even for fixed values of r. The results utilize reductions from the DOMINATING SET problem on 3-regular graphs (Kikuno, Yoshida, and Kakuda 1980; Garey and Johnson 1979) and the problem of partitioning a K_4-free 4-regular graph into triangles (van Rooij, van Kooten Niekerk, and Bodlaender 2013), respectively.

Theorem 15. IN-CLUSTERING *is* NP*-complete for* $r = 4$*, and* DIAM-CLUSTERING *is* NP*-complete for* $r = 6$*.*

Having ruled out fixed-parameter tractability when parameterizing only by r, we turn to the case where the parameter is k alone. First of all, for $k = 1$ ANY-CLUSTERING is equivalent to CLOSEST STRING, a well-studied NP-complete problem (Gramm, Niedermeier, and Rossmanith 2003). Using a two-step reduction from 3-COLORING on 4-regular graphs (Dailey 1980), we show that DIAM-CLUSTERING is also NP-complete, even when restricted to a fixed value of k.

Theorem 16. DIAM-CLUSTERING *is* NP*-complete for* $k = 3$*.*

Unlike the previous two problems, IN-CLUSTERING admits a simple polynomial-time brute-force algorithm for every fixed value of k where the order of the polynomial depends on k (*i.e.*, the problem is in XP). However, we can still exclude fixed-parameter tractability:

Theorem 17. IN-CLUSTERING *is* W[2]*-complete parameterized by* k *and can be solved in time* $\mathcal{O}(|M|^k|M|kd)$*. Moreover, there is no algorithm solving* IN-CLUSTERING *in time* $|M|^{o(k)}$ *unless the Exponential Time Hypothesis fails.*

The above results already show that out of the three considered parameters, k and r must both be used if one wishes to obtain fixed-parameter algorithms for the clustering problems under consideration. In the case of clustering of incomplete data, the only two questions that remain are whether one also needs to use the covering number, and whether it is possible to extend the polynomial-time algorithm for IN-CLUSTERING to IN-CLUSTERING-COMPLETION. We resolve these last questions using reductions from 3-COLORING and CLOSEST STRING.

Theorem 18. *For* $X \in \{$IN,ANY,DIAM$\}$*,* X-CLUSTERING-COMPLETION *is* NP*-complete even if* $k = 3$ *and* $r = 0$*. Furthermore,* IN-CLUSTERING-COMPLETION *is* NP*-complete even if* $k = 1$ *and there is only one row containing* $\square$*-entries.*

Going Beyond Boolean Domain

In this section, we briefly discuss two generalizations of the clustering problems under consideration that allow for larger domain size, where each generalization is based on a different way of measuring distance between vectors in higher domains. In particular, we discuss the Hamming distance and the Manhattan distance over a domain $Q = \{0, 1, \ldots, q-1, \square\}$, for some $q \geq 2$. Our aim in this section is to extend our results from matrices over the Boolean domain to these generalizations, and the main tools we use are two encodings of domain values. We define the two encodings $\alpha : [q] \cup \{\square\} \to \{0,1,\square\}^q$ and $\beta : [q] \cup \{\square\} \to \{0,1,\square\}^q$, where $\alpha(i)$ is the binary encoding of 2^i and $\beta(i)$ is the unary encoding of i if $i \neq \square$ and $\alpha(i) = \beta(i) = \square^q$, otherwise. Moreover, for $\vec{v} \in \{0,1\}^d$, we let $\alpha(\vec{v})$ and $\beta(\vec{v})$ be the vectors in $\{0,1\}^{qd}$ obtained from $\vec{v}$ by replacing each coordinate $i \in [d]$ with a *block* of q coordinates equal to $\alpha(i)$ and $\beta(i)$, respectively. For example, if $Q = \{0,1,2,\square\}$ and $d = 2$, then $\alpha((0,2)) = (0,0,1,1,0,0)$ and $\beta((0,2)) = (0,0,0,0,1,1)$.

It is easy to verify that there is a direct correspondence between the vector distances in a matrix M over Q^d and the Hamming vector distances in the matrix over $\{0,1,\square\}^{qd}$ obtained by applying the respective encoding function to M.

Observation 19. *For each* $\vec{a}, \vec{b} \in Q^d$ *it holds that* $\delta(\vec{a},\vec{b}) \cdot 2 = \delta(\alpha(\vec{a}),\alpha(\vec{b}))$ *and that* $\sum_{t=1}^{d} |a[t] - b[t]| = \delta(\beta(\vec{a}),\beta(\vec{b}))$*.*

Consider a matrix M obtained by applying α (or β) to a matrix M'. A completion M^* of M is *block-preserving* w.r.t. α (respectively β) if for each vector $\vec{v} \in M^*$ the i-th block of $\vec{v}$ is equal to $\alpha(i)$ (respectively $\beta(i)$) for some $i \in Q$. Equivalently, M^* is block-preserving w.r.t. α (or β) if it can be obtained by applying α (or β, respectively) to the elements of some completion of the matrix M'.

For PROB $\in$ IN/ANY/DIAM-CLUSTERING-COMPLETION, let PROB$_\alpha$ and PROB$_\beta$ be the adaptation of PROB to the case where we additionally require the completion M^* of M to be block-preserving (w.r.t. α or β). Since both encodings only increase the dimension of the vectors by a constant factor, Observation 19 allows us to reduce the completion problems over Q to the question of finding block-preserving completions of Boolean matrices. It is easy to argue that all the developed algorithmic techniques can be extended to the block-preserving variants of the problems. As a corollary, we obtain that all our FPT-results and XP-results also carry over to the finite domain case.

Conclusion

We provided a systematic study of the parameterized complexity of fundamental clustering problems for incomplete data. Our results draw a detailed map of the complexity landscape for the studied problems and showcase a sharp contrast between the settings that are fixed-parameter tractable and those which are not.

Finally, we believe that the insights and techniques showcased in this paper are of general interest. Indeed, in essence they show that vectors over a bounded domain which are packed in dense clusters have non-trivial combinatorial properties that only become accessible through a suitable set representation. We hope that these insights and techniques turn out to be useful in other settings as well.

Acknowledgements

Robert Ganian acknowledges support by the Austrian Science Fund (FWF, projects P31336 and Y 1329). Stefan Szeider acknowledges support by the Austrian Science Fund (FWF, project P32441) and the Vienna Science and Technology Fund (WWTF, project ICT19-065). Sebastian Ordyniak acknowledges support from the Engineering and Physical Sciences Research Council (EPSRC, project EP/V00252X/1).

References

Aggarwal, C. C.; and Reddy, C. K. 2013. *Data Clustering: Algorithms and Applications*. Chapman & Hall/CRC, 1st edition.

Balzano, L.; Szlam, A.; Recht, B.; and Nowak, R. D. 2012. K-subspaces with missing data. *2012 IEEE Statistical Signal Processing Workshop (SSP)* 612–615.

Boucher, C.; and Ma, B. 2011. Closest String with Outliers. *BMC Bioinformatics* 12(S-1): S55.

Cabello, S.; Giannopoulos, P.; Knauer, C.; Marx, D.; and Rote, G. 2011. Geometric clustering: Fixed-parameter tractability and lower bounds with respect to the dimension. *ACM Trans. Algorithms* 7(4): 43:1–43:27.

Candès, E. J.; and Plan, Y. 2010. Matrix Completion With Noise. *Proceedings of the IEEE* 98(6): 925–936.

Candès, E. J.; and Recht, B. 2009. Exact Matrix Completion via Convex Optimization. *Foundations of Computational Mathematics* 9(6): 717–772.

Candès, E. J.; and Tao, T. 2010. The power of convex relaxation: near-optimal matrix completion. *IEEE Trans. Information Theory* 56(5): 2053–2080.

Charikar, M.; and Panigrahy, R. 2004. Clustering to minimize the sum of cluster diameters. *Journal of Computer and System Sciences* 68(2): 417 – 441.

Chen, J.; Hermelin, D.; and Sorge, M. 2019. On Computing Centroids According to the p-Norms of Hamming Distance Vectors. In Bender, M. A.; Svensson, O.; and Herman, G., eds., *27th Annual European Symposium on Algorithms, ESA 2019, September 9-11, 2019, Munich/Garching, Germany*, volume 144 of *LIPIcs*, 28:1–28:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik.

Cygan, M.; Fomin, F. V.; Kowalik, L.; Lokshtanov, D.; Marx, D.; Pilipczuk, M.; Pilipczuk, M.; and Saurabh, S. 2015. *Parameterized Algorithms*. Springer.

Dailey, D. P. 1980. Uniqueness of colorability and colorability of planar 4-regular graphs are NP-complete. *Discrete Mathematics* 30(3): 289 – 293.

Downey, R. G.; and Fellows, M. R. 2013. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer. ISBN 978-1-4471-5558-4, 978-1-4471-5559-1.

Dyer, M.; and Frieze, A. 1985. A Simple Heuristic for the p-centre Problem. *Oper. Res. Lett.* 3(6): 285–288.

Elhamifar, E. 2016. High-Rank Matrix Completion and Clustering under Self-Expressive Models. In Lee, D. D.; Sugiyama, M.; Luxburg, U. V.; Guyon, I.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 29*, 73–81. Curran Associates, Inc.

Elhamifar, E.; and Vidal, R. 2013. Sparse Subspace Clustering: Algorithm, Theory, and Applications. *IEEE Trans. Pattern Anal. Mach. Intell.* 35(11): 2765–2781.

Erdös, P.; and Rado, R. 1960. Intersection theorems for systems of sets. *Journal of the London Mathematical Society* 1(1): 85–90.

Feder, T.; and Greene, D. 1988. Optimal Algorithms for Approximate Clustering. In *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing*, STOC '88, 434–444. ACM.

Flum, J.; and Grohe, M. 2006. *Parameterized Complexity Theory*, volume XIV of *Texts in Theoretical Computer Science. An EATCS Series*. Berlin: Springer.

Fomin, F. V.; Golovach, P. A.; Lokshtanov, D.; Panolan, F.; and Saurabh, S. 2020. Approximation Schemes for Low-rank Binary Matrix Approximation Problems. *ACM Trans. Algorithms* 16(1): 12:1–12:39.

Fomin, F. V.; Golovach, P. A.; and Panolan, F. 2020. Parameterized low-rank binary matrix approximation. *Data Min. Knowl. Discov.* 34(2): 478–532. doi:10.1007/s10618-019-00669-5. URL https://doi.org/10.1007/s10618-019-00669-5.

Fomin, F. V.; Golovach, P. A.; and Simonov, K. 2019. Parameterized k-Clustering: Tractability Island. In Chattopadhyay, A.; and Gastin, P., eds., *39th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2019, December 11-13, 2019, Bombay, India*, volume 150 of *LIPIcs*, 14:1–14:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik.

Fomin, F. V.; Lokshtanov, D.; Saurabh, S.; and Zehavi, M. 2019. *Kernelization: Theory of Parameterized Preprocessing*. Cambridge University Press. doi:10.1017/9781107415157.

Frances, M.; and Litman, A. 1997. On covering problems of codes. *Theory of Computing Systems* 30(2): 113–119.

Gan, G.; Ma, C.; and Wu, J. 2007. *Data clustering - theory, algorithms, and applications.* SIAM.

Ganian, R.; Kanj, I.; Ordyniak, S.; and Szeider, S. 2018. Parameterized Algorithms for the Matrix Completion Problem. In *ICML*, volume 80 of *JMLR Workshop and Conference Proceedings*, 1642–1651.

Garey, M.; and Johnson, D. 1979. *Computers and Intractability*. W.H. Freeman.

Gaspers, S.; and Szeider, S. 2014. Guarantees and limits of preprocessing in constraint satisfaction and reasoning. *Artificial Intelligence* 216: 1–19.

Gąsieniec, L.; Jansson, J.; and Lingas, A. 1999. Efficient Approximation Algorithms for the Hamming Center Problem. In *Proceedings of the Tenth Annual ACM-SIAM Symposium on Discrete Algorithms*, 905–906.

Gąsieniec, L.; Jansson, J.; and Lingas, A. 2004. Approximation algorithms for Hamming clustering problems. *Journal of Discrete Algorithms* 2(2): 289 – 301.

Gonzalez, T. F. 1985. Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science* 38: 293 – 306.

Gottlob, G.; and Szeider, S. 2008. Fixed-parameter algorithms for artificial intelligence, constraint satisfaction, and database problems. *The Computer Journal* 51(3): 303–325. Survey paper.

Gramm, J.; Niedermeier, R.; and Rossmanith, P. 2003. Fixed-Parameter Algorithms for CLOSEST STRING and Related Problems. *Algorithmica* 37(1): 25–42.

Hardt, M.; Meka, R.; Raghavendra, P.; and Weitz, B. 2014. Computational Limits for Matrix Completion. In *Proceedings of The 27th Conference on Learning Theory*, volume 35 of *JMLR Workshop and Conference Proceedings*, 703–725. JMLR.org.

Hermelin, D.; and Rozenberg, L. 2015. Parameterized complexity analysis for the Closest String with Wildcards problem. *Theoretical Computer Science* 600: 11–18.

Jiao, Y.; Xu, J.; and Li, M. 2004. On the k-Closest Substring and k-Consensus Pattern Problems. In Sahinalp, S. C.; Muthukrishnan, S.; and Dogrusöz, U., eds., *Combinatorial Pattern Matching, 15th Annual Symposium, CPM 2004, Istanbul,Turkey, July 5-7, 2004, Proceedings*, volume 3109 of *Lecture Notes in Computer Science*, 130–144. Springer.

Kikuno, T.; Yoshida, N.; and Kakuda, Y. 1980. The NP-Completeness of the dominating set problem in cubic planer graphs. *IEICE TRANSACTIONS (1976-1990)* 63(6): 443–444.

Koana, T.; Froese, V.; and Niedermeier, R. 2020a. Complexity of Combinatorial Matrix Completion With Diameter Constraints. *CoRR* abs/2002.05068.

Koana, T.; Froese, V.; and Niedermeier, R. 2020b. Parameterized Algorithms for Matrix Completion with Radius Constraints. In Gørtz, I. L.; and Weimann, O., eds., *31st Annual Symposium on Combinatorial Pattern Matching, CPM 2020, June 17-19, 2020, Copenhagen, Denmark*, volume 161 of *LIPIcs*, 20:1–20:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik.

Kratsch, S.; Marx, D.; and Wahlström, M. 2016. Parameterized Complexity and Kernelizability of Max Ones and Exact Ones Problems. *TOCT* 8(1): 1:1–1:28.

Leskovec, J.; Rajaraman, A.; and Ullman, J. D. 2014. *Mining of Massive Datasets*. New York, NY, USA: Cambridge University Press, 2nd edition.

Marx, D. 2005. Parameterized complexity of constraint satisfaction problems. *Computational Complexity* 14(2): 153–183.

Mirkin, B. 2005. *Clustering For Data Mining: A Data Recovery Approach*. Chapman & Hall/CRC.

van Rooij, J. M. M.; van Kooten Niekerk, M. E.; and Bodlaender, H. L. 2013. Partition Into Triangles on Bounded Degree Graphs. *Theory of Computing Systems* 52(4): 687–718.

Yi, J.; Yang, T.; Jin, R.; Jain, A. K.; and Mahdavi, M. 2012. Robust Ensemble Clustering by Matrix Completion. In *2012 IEEE 12th International Conference on Data Mining*, 1176–1181.